

# Isaac Cheung

cheungis.github.io  
isaac.cheung@hotmail.com | 604.500.3129

## EDUCATION

**UNIVERSITY OF  
BRITISH COLUMBIA**  
BSc IN COMPUTER SCIENCE  
3RD YEAR STANDING  
Vancouver, BC  
September 2018 - April 2021

GPA: 3.92 / 4.33  
Major GPA: 4.03 / 4.33

## LINKS

GitHub:// [github.com/cheungis](https://github.com/cheungis)  
LinkedIn:// [linkedin.com/in/cheungis](https://linkedin.com/in/cheungis)

## COURSEWORK

**UNDERGRADUATE**  
Computer Systems  
Data Structures and Algorithms  
Software Construction  
Formal Systems and Logic  
Foundations of Computing

## SKILLS

**LANGUAGES**  
Python • Java • C • C++ • Matlab  
JavaScript • HTML •  $\LaTeX$

**VERSION CONTROL SYSTEMS**  
GitHub

**TESTING**  
JUnit

**FRAMEWORKS / ETC**  
Node.js • Swing • Pygame • CSS

## AWARDS

**HONOR ROLL**  
Sept 2012 - June 2017

**WATERLOO CEMC**  
CERTIFICATE OF DISTINCTION  
Feb 2015

**BC ACHIEVEMENT SCHOLARSHIP**  
Sept 2017

## PROJECTS

### SOUNDBOARD | JULY 2019

- Implemented a soundboard app using Android studio.
- Utilizes event listeners implemented with the observer design pattern.
- Created a desktop version using Java Swing.

### DISCORD BOTS | DECEMBER 2018

- Developed 2 Discord Bots with JavaScript and the Discord API.
- Utilizes Discord JS, a node.js module.
- Designed with best practices in mind, such as dynamic command handling.
- Bot #1 generates links to allow for ease of access to websites.
- Bot #2 automates the process of mass deleting server messages.
- Bot #2 allows for the option of searching and filtering messages to include or exclude attributes, such as the message author.

### LEAGUE OF LEGENDS PROFILE ANALYZER | SEPTEMBER 2018

- Extracts data from players and store them in profiles to analyze and compare with one another with Java.
- Incorporated design patterns to solve problems encountered during development including the iterator and observer design patterns.
- Unit testing was done on all methods to ensure correctness of code.
- GUI built with Java Swing.

### 2048 GAME | FEBRUARY 2018

- Implemented the 2048 game in Python, a popular puzzle game originally released in 2014.
- Utilizes personally developed algorithm to resolve collapsing rows.
- Black box and unit testing was done on all methods to ensure correctness of code.

### MAZE SOLVER PROGRAM | FEBRUARY 2018

- Constructed a program that can determine whether or not a maze is solvable using Python.
- Solves the maze via a depth first search algorithm.

### ENCRYPTER AND DECRYPTER | JANUARY 2018

- Assembled a program to encrypt and decrypt messages with Python.
- Employs encryption techniques from trans positional ciphers.
- Added functionality to decrypt messages via brute force approach.

### CLASSIC GAMES | OCTOBER 2017

- Recreated classic games using Object Oriented Programming in Python.
- Games include pong, tic tac toe and a memory game.
- Designed with OOP and OOD paradigms, simplifying modification and expansion of content for the games that were recreated.
- GUI Designed and implemented through the pygame library.